Improving Document-Level Sentiment Analysis with User and Product Context

Chenyang Lyu[1], Jennifer Foster[1], Yvette Graham[2]

[1]School of Computing, Dublin City University, Ireland

[2]School of Computer Science & Statistics, Trinity College Dublin, Ireland

COLING 2020

Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

DCU
Ollscoil Chathair Bhaile Átha Cliath
Dublin City University

Science Foundation Ireland
sfi
For what's next

Outline

1. Background
2. Methodology
3. Experiments and Analysis
4. Conclusion

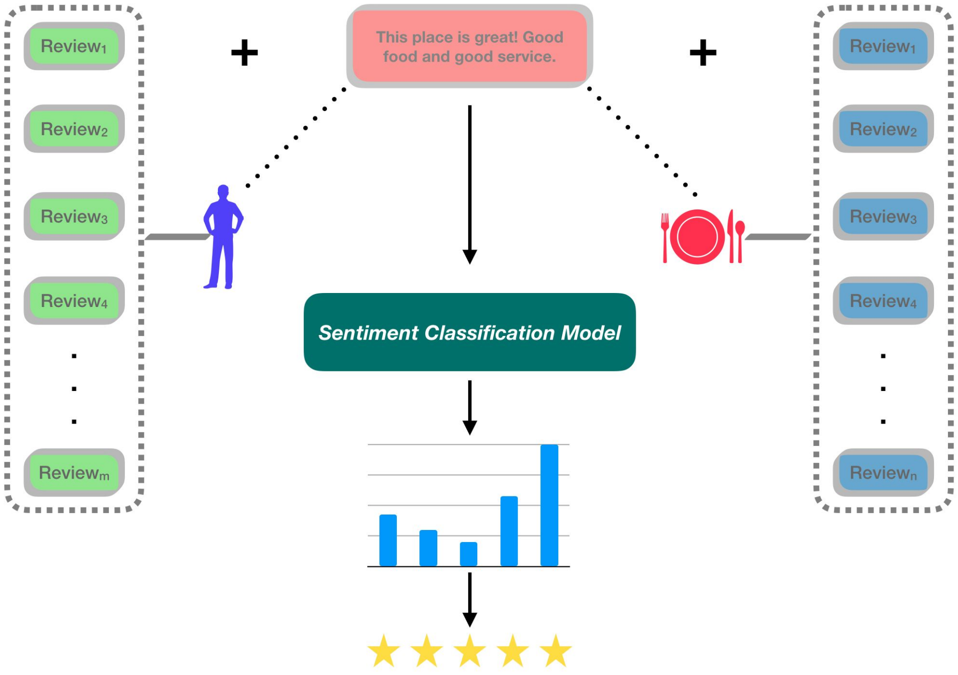
1. Background

- Document-Level Sentiment Analysis
  - Classify the sentiment (positive or negative) expressed by a whole document.
  - Generally the sentiment is on a single entity.
- Sentiment Analysis with User and Product Information
  - Not only we have the review text, but also the user and product IDs.
  - Modeling the user, who has written the review, and the product being reviewed is worthwhile for polarity prediction.

1. Background

- Previous methods:
  - Modeling each user and product as embedding vectors.
  - Implicitly learning user and product preferences during training.
- Challenges:
  - User and product embeddings purely learned from training are not well enough to capture user and product preferences.
  - How to explicitly take advantage of historical reviews associated with given user and product.

1. Background



But, how?

Possible solution:

- Compute representations of all historical reviews associated with a certain user and product during training.

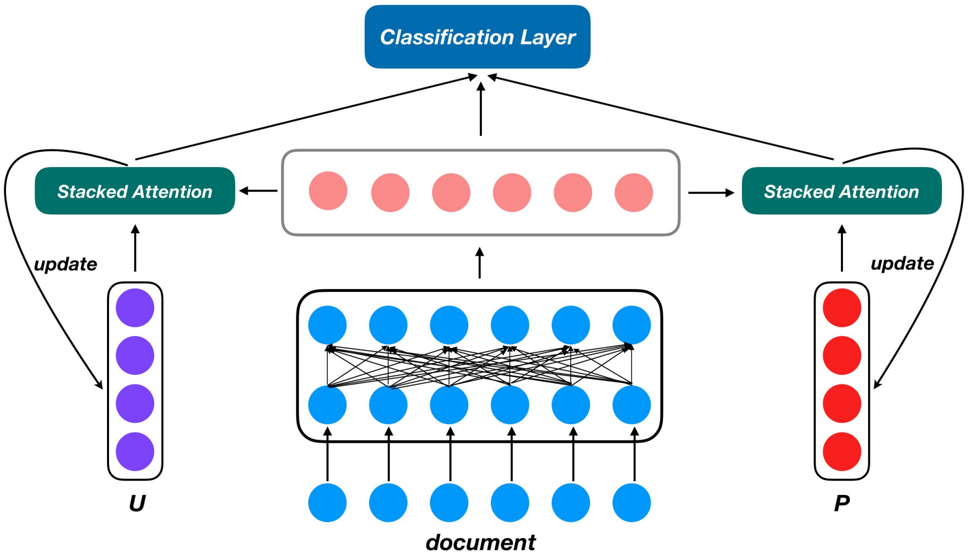
  - time-consuming

- Pre-compute representations of all reviews using a pretrained model, then use them during training.

  - memory-consuming

2. Methodology

Incorporating User and Product Context



Step 1:

- Obtain document representation.

Step 2:

- Use user and product embedding vectors to gather information from document representation through attention function.

Step 3:

- Fuse user-biased and product-biased information to obtain a final review representation, then pass it to a classification layer to get sentiment label.

Step 4:

- Incrementally add current biased representation to corresponding user and product embeddings.

2. Methodology - Incorporating User and Product Context

Get document representation:

$$H_d = BERT\_encoder(d) \; (1)$$

Inject user and product preferences:

$$C_u^t = stacked - attention(E_u, H_d) \; (2)$$

$$C_p^t = stacked - attention(E_p, H_d) \; (3)$$

Gating mechanism:

$$z_u = \sigma(W_{zu} C_u^t + W_{zh} H_d + b_u) \; (4)$$

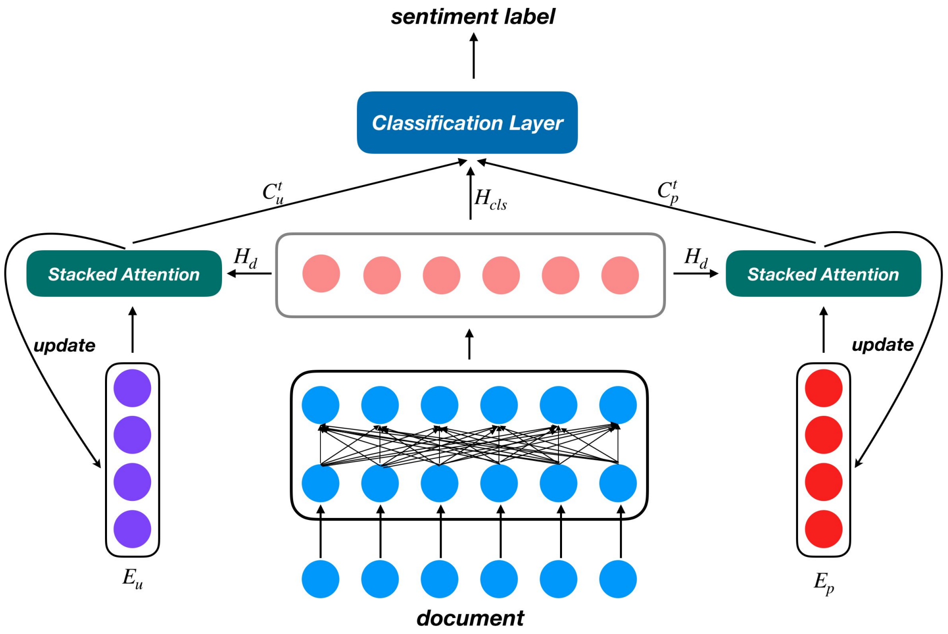
$$z_p = \sigma(W_{zp} C_p^t + W_{zh} H_d + b_p) \; (5)$$

Final representation:

$$H_{biased} = H_{cls} + z_u \odot C_u^t + z_p \odot C_p^t \; (6)$$

Update user and product matrix:

$$E_u^{'} = \sigma(E_u + \lambda_u C_u^t) \quad E_p^{'} = \sigma(E_p + \lambda_p C_p^t)(7)$$



3. Experiments and Analysis

- Datasets:
  - Our experiments are conducted on the IMDB, Yelp-13 and Yelp-14 benchmark datasets.

Datasets	Classes	Documents	Users	Products	Docs/User	Docs/Product	Words/Doc
IMDB	1–10	84,919	1,310	1,635	64.82	51.94	394.6
Yelp-2013	1–5	78,966	1,631	1,633	48.42	48.36	189.3
Yelp-2014	1–5	231,163	4,818	4,194	47.97	55.11	196.9

Table 1: Statistics of IMDB, Yelp-2013 and Yelp-2014.

- Experimental setup:
  - Learning rate: {8e-6, 3e-5, 5e-5}, weight decay: {0, 1e-1, 1e-2, 1e-3}
  - Warmup ratio: 0.1, linear decay.
  - Maximum length to BERT: 512 wordpiece tokens.
  - Optimizer: AdamW.
  - Batch size:{8, 16}
  - Epochs:{2, 3}

3. Experiments and Analysis

- Experimental results:
  - Our proposed model achieves the best accuracy and RMSE on Yelp-2013 and Yelp-2014, and the best RMSE on IMDB.

	IMDB		Yelp-2013		Yelp-2014	
	Acc. (%)	RMSE	Acc. (%)	RMSE	Acc. (%)	RMSE
BERT VANILLA	$47.9_{0.46}$	$1.243_{0.019}$	$67.2_{0.46}$	$0.647_{0.011}$	$67.5_{0.71}$	$0.621_{0.012}$
IUPC W/O UPDATE	$52.1_{0.31}$	$1.194_{0.010}$	$69.7_{0.37}$	$0.605_{0.007}$	$70.0_{0.29}$	$0.601_{0.007}$
IUPC (our model)	$53.8_{0.57}$	$\mathbf{1.151_{0.013}}$	$\mathbf{70.5_{0.29}}$	$\mathbf{0.589_{0.004}}$	$\mathbf{71.2_{0.26}}$	$\mathbf{0.592_{0.008}}$
UPNN	43.5	1.602	59.6	0.784	60.8	0.764
UPDMN	46.5	1.351	63.9	0.662	61.3	0.720
NSC	53.3	1.281	65.0	0.692	66.7	0.654
CMA	54.0	1.191	66.3	0.677	67.6	0.637
DUPMN	53.9	1.279	66.2	0.667	67.6	0.639
HCSC	54.2	1.213	65.7	0.660	67.6	0.639
HUAPA	55.0	1.185	68.3	0.628	68.6	0.626
CHIM	**56.4**	1.161	67.8	0.641	69.2	0.622
RRP-UPM	56.2	1.174	69.0	0.629	69.1	0.621

3. Experiments and Analysis

- Low-resource analysis:
  - We select only reviews where the number of reviews by that user or for that product falls below three thresholds: 40%, 60%, 80%, where % stands for the number of reviews for a given user/product relative to the average number of reviews for all users/products.
  - Our proposed model achieves better accuracy and RMSE when there are only a small number of previous reviews available for a given product/user.

	40%		60%		80%	
	Acc. (%)	RMSE	Acc. (%)	RMSE	Acc. (%)	RMSE
IUPC W/O UPDATE	63.0	0.608	64.0	0.665	66.8	0.643
IUPC (our model)	**65.7**	**0.585**	**66.8**	**0.649**	**67.9**	**0.631**

Table 3: Analysis of three lower-resource scenarios where % denotes a threshold filter corresponding to the proportion of reviews available relative to the average number in the dataset Yelp-2013 (dev).

4. Conclusion

- In this paper, we propose a neural sentiment analysis architecture that explicitly utilizes all past reviews from a given user or product to improve sentiment polarity classification on the document level.
- Our experimental results on the IMDB, Yelp-13 and Yelp-14 datasets demonstrate that incorporating this additional context is effective, particularly for the Yelp datasets.

Thanks for listening!